| Notice of Allowability | Application No. 09/715,428 | Applicant(s) BENTZ, OLE | |
|---|---|---|---|
| | Examiner Jin-Cheng Wang | Art Unit 2672 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--***

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *7/25/2005 and 11/18/2005*.
2. ☒ The allowed claim(s) is/are *10,17-20,26 and 28-39*.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review ( PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

**Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).**

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

*EXAMINER'S AMENDMENT*

An examiner's amendment to the record appears below. Should the changes and or additions be unacceptable to the applicants, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Prior to this Office Action, the Examiner has an interview with Mr. Kimton N. Eng dated December 15, 2005. Applicant agreed with the changes suggested by the Examiner to the claims 10, 36 and 38.

10. (Currently Amended) A method of calculating a texture coordinate for a texture map having a size from an input texture coordinate value located in one of a plurality of predefined input ranges, comprising:

~~concurrently~~ calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [input + (1 * tex_size)] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [input – (1 * tex_size)] and a second value ~~B = [input + (2 * tex_size)],~~ <u>B = [input – (2 * tex_size)]</u>, input is the input texture coordinate value and tex_size is the size of the texture map;

<u>concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value;</u> and

selecting an output texture coordinate from the plurality of ~~concurrently~~ calculated texture coordinate values and the input texture coordinate value, where the sign of the input

texture coordinate value is negative, selecting B when (A < 0), otherwise selecting A as the corresponding texture coordinate, and where the input texture coordinate value is equal to zero or the sign of the input texture coordinate is positive, selecting the input texture coordinate value when (A < 0), selecting A when (B < 0), and selecting B otherwise.

36. (Currently Amended) A method of calculating a texture coordinate for a texture map having a size from an input texture coordinate value located in one of a plurality of predefined input ranges, comprising:

~~concurrently~~ calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [input + (1 * tex_size)] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [input – (1 * tex_size)] and a second value ~~B = [input + (2 * tex_size)],~~ B = [input – (2 * tex_size)], and further calculating a third value C = (tex_size – A), input is the input texture coordinate value and tex_size is the size of the texture map;

concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value; and

selecting an output texture coordinate from the plurality of ~~concurrently~~ calculated texture coordinate values and the input texture coordinate value, where the sign of the input texture coordinate value is negative, selecting B when (A < 0), otherwise selecting C as the corresponding texture coordinate, and where the input texture coordinate is equal to zero or the sign of the input texture coordinate is positive, selecting the input texture coordinate value when

(A < 0), selecting C when (B < 0), and selecting B otherwise.

38. (Currently Amended) A method of calculating a texture coordinate for a texture map having a size from an input texture coordinate value located in one of a plurality of predefined input ranges, comprising:

~~concurrently~~ calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [2 * tex_size] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [– 2* tex_size + 1LSB] and a second value B = [input – (2 * tex_size)], and further calculating a third value C = (0 – A), input is the input texture coordinate value, tex_size is the size of the texture map, and 1LSB is a binary value equal to 1 and having a bit length the same as the tex_size;

<u>concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value;</u> and

selecting an output texture coordinate from the plurality of ~~concurrently~~ calculated texture coordinate values and the input texture coordinate value, where the sign of the second value B is negative, selecting C as the output texture coordinate, and otherwise, selecting the input texture coordinate as the output texture coordinate; and

clamping the selected output texture coordinate to a clamped value.

***Reasons for Allowance***

The following is an examiner's statement of reasons for allowance:

Nothing in the prior art anticipates or suggests, “calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [input + (1 * tex_size)] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [input – (1 * tex_size)] and a second value <u>B = [input – (2 * tex_size)]</u>, input is the input texture coordinate value and tex_size is the size of the texture map;

concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value; and

selecting an output texture coordinate from the plurality of calculated texture coordinate values and the input texture coordinate value, where the sign of the input texture coordinate value is negative, selecting B when (A < 0), otherwise selecting A as the corresponding texture coordinate, and where the input texture coordinate value is equal to zero or the sign of the input texture coordinate is positive, selecting the input texture coordinate value when (A < 0), selecting A when (B < 0), and selecting B otherwise” as set forth in the independent Claim 10.

The Claim 10 has been amended by the Examiner in present Office Action to correct errors in the claim language set forth in the applicant's previously submitted claim 10 of 7/15/2005. The Claim 10 as currently amended would accommodate a graphics processing system to perform texture processing in a single device with the addressing mode operations for arbitrary sized texture maps. The claim 20 depends upon the claim 10 and is allowed for the same reasons given for the independent claim 10.

The prior art Grossman et al reference discloses the use of texture clamping, the texture addressing circuit, and the processing logic. Grossman further discloses linear interpolation of

texture coordinates and the field definitions within an input texture map coordinate that supports clamping and a multi-pass technique for tiling large texture maps wherein an field is used to **expand** the address space of textures **beyond** the zero to one coordinate range stored in a hardware texture map. Grossman further discloses in the processing logic that the compare value obtained from a compare register is tested against the masked value produced in a processing block and the result of this test determines whether or not an input coordinate is within a particular texture coordinate range in which texturing is enabled. Grossman does not teach the specific formula of calculating texture coordinates together with the specific way of selecting texture coordinates for texture remapping or clamping so that out-of-range texture coordinates can be re-mapped to the range of a texture map including the border of the range of the texture map. Grossman failed to teach the formula used in selecting the corresponding texture coordinates for a mapping mode.

Nothing in the prior art anticipates or suggests, "a plurality of coordinate calculation circuits corresponding to a plurality of input coordinate ranges defined outside of the acceptable range for both negative and positive input coordinate values, each coordinate calculation circuit coupled to receive a signal corresponding to the sign of the input coordinate value and a respective texture size value corresponding to a multiple of the size of the texture map, each coordinate calculation circuit providing a respective signed coordinate output value calculated from the input texture coordinate value and the size of the texture map, a first and a second coordinate calculation circuit of the plurality including a negating circuit configured to generate an output value corresponding to a positive or negative respective texture size value in

accordance with the sign of the input coordinate value, and further including a summing circuit having a first input coupled to negating circuit and a second input at which a second input value is provided, the summing circuit configured to generate an output corresponding to the sum of the output value of the negating circuit and a value received by at the second input; a selection circuit coupled to concurrently receive as input values the input coordinate and the signed coordinate output values of the plurality of coordinate calculation circuits, the selection circuit selecting one of the input values as an output texture coordinate value; and select logic coupled to the selection circuit and further coupled to receive input signals corresponding to the sign of the input coordinate value and the signs of the coordinate output values, the select logic providing a selection signal commanding the selection circuit to select one of the input values as the output texture coordinate in accordance with the received input signals" in a texture addressing circuit for calculating texture coordinates for a texture map having a size and an acceptable range of input coordinate values set forth in the independent claim 26.

The Claim 26 in the amended claimed invention of 07/25/2005 would accommodate a graphics processing system to perform texture processing in a single device with the addressing mode operations for arbitrary sized texture maps. The claims 28-35 depend upon the claim 26 and are allowed for the same reasons set forth in above for the independent claim 26.

The prior art Grossman et al reference discloses the use of texture clamping, the texture addressing circuit, and the processing logic. Grossman further discloses linear interpolation of texture coordinates and the field definitions within an input texture map coordinate that supports clamping and a multi-pass technique for tiling large texture maps wherein an field is used to **expand** the address space of textures **beyond** the zero to one coordinate range stored in a

hardware texture map. Grossman further discloses in the processing logic that the compare value obtained from a compare register is tested against the masked value produced in a processing block and the result of this test determines whether or not an input coordinate is within a particular texture coordinate range in which texturing is enabled. Grossman does not teach the specific formula of calculating texture coordinates together with the specific way of selecting texture coordinates for texture remapping or clamping so that out-of-range texture coordinates can be re-mapped to the range of a texture map including the border of the range of the texture map. Grossman failed to teach the formula used in selecting the corresponding texture coordinates for a mapping mode.

Nothing in the prior art anticipates or suggests, "calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [input + (1 * tex_size)] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [input – (1 * tex_size)] and a second value B = [input – (2 * tex_size)], and further calculating a third value C = (tex_size – A), input is the input texture coordinate value and tex_size is the size of the texture map;

concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value; and

selecting an output texture coordinate from the plurality of calculated texture coordinate values and the input texture coordinate value, where the sign of the input texture coordinate value is negative, selecting B when (A < 0), otherwise selecting C as the corresponding texture

coordinate, and where the input texture coordinate is equal to zero or the sign of the input texture coordinate is positive, selecting the input texture coordinate value when (A < 0), selecting C when (B < 0), and selecting B otherwise" in a method of calculating a texture coordinate for a texture map having a size from an input texture coordinate value located in one of a plurality of predefined input ranges set forth in the independent claim 36.

The Claim 36 has been amended by the Examiner in present Office Action to correct errors in the claim language set forth in the applicant's previously submitted claim 36 of 7/15/2005. The Claim 36 as currently amended would accommodate a graphics processing system to perform texture processing in a single device with the addressing mode operations for arbitrary sized texture maps. The claim 37 depends upon the claim 36 and is allowed for the same reasons given for the independent claim 36.

The prior art Grossman et al reference discloses the use of texture clamping, the texture addressing circuit, and the processing logic. Grossman further discloses linear interpolation of texture coordinates and the field definitions within an input texture map coordinate that supports clamping and a multi-pass technique for tiling large texture maps wherein an field is used to **expand** the address space of textures **beyond** the zero to one coordinate range stored in a hardware texture map. Grossman further discloses in the processing logic that the compare value obtained from a compare register is tested against the masked value produced in a processing block and the result of this test determines whether or not an input coordinate is within a particular texture coordinate range in which texturing is enabled. Grossman does not teach the specific formula of calculating texture coordinates together with the specific way of selecting

texture coordinates for texture remapping or clamping so that out-of-range texture coordinates can be re-mapped to the range of a texture map including the border of the range of the texture map. Grossman failed to teach the formula used in selecting the corresponding texture coordinates for a mapping mode.

Nothing in the prior art anticipates or suggests, "calculating a plurality of signed texture coordinate values corresponding to the plurality of predefined input coordinate ranges, where the sign of the input texture coordinate value is negative, calculating a first value A = [2 * tex_size] and a second value B = [input + (2 * tex_size)], and otherwise, calculating a first value A = [– 2* tex_size + 1LSB] and a second value B = [input – (2 * tex_size)], and further calculating a third value C = (0 – A), input is the input texture coordinate value, tex_size is the size of the texture map, and 1LSB is a binary value equal to 1 and having a bit length the same as the tex_size;

concurrently receiving the plurality of signed texture coordinate values and the input texture coordinate value; and

selecting an output texture coordinate from the plurality of ~~concurrently~~ calculated texture coordinate values and the input texture coordinate value, where the sign of the second value B is negative, selecting C as the output texture coordinate, and otherwise, selecting the input texture coordinate as the output texture coordinate; and

clamping the selected output texture coordinate to a clamped value" in a method of calculating a texture coordinate for a texture map having a size from an input texture coordinate value located in one of a plurality of predefined input ranges set forth in the independent claim 36.

The Claim 38 has been amended by the Examiner in present Office Action to correct errors in the claim language set forth in the applicant's previously submitted claim 38 of 7/15/2005. The Claim 38 as currently amended would accommodate a graphics processing system to perform texture processing in a single device with the addressing mode operations for arbitrary sized texture maps. The claim 17-19 and 39 depend upon the claim 38 and are allowed for the same reasons given for the independent claim 38.

The prior art Grossman et al reference discloses the use of texture clamping, the texture addressing circuit, and the processing logic. Grossman further discloses linear interpolation of texture coordinates and the field definitions within an input texture map coordinate that supports clamping and a multi-pass technique for tiling large texture maps wherein an field is used to **expand** the address space of textures **beyond** the zero to one coordinate range stored in a hardware texture map. Grossman further discloses in the processing logic that the compare value obtained from a compare register is tested against the masked value produced in a processing block and the result of this test determines whether or not an input coordinate is within a particular texture coordinate range in which texturing is enabled. Grossman does not teach the specific formula of calculating texture coordinates together with the specific way of selecting texture coordinates for texture remapping or clamping so that out-of-range texture coordinates can be re-mapped to the range of a texture map including the border of the range of the texture map. Grossman failed to teach the formula used in selecting the corresponding texture coordinates for a mapping mode.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jin-Cheng Wang whose telephone number is (571) 272-7665. The examiner can normally be reached on 8:00 - 6:30 (Mon-Thu).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mike Razavi can be reached on (571) 272-7664. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jcw

Jeffery A. Brier
PRIMARY EXAMINER